

УДК 004.453/.453: 004.451.9Linux
ББК 32.972.1
С37

Симмондс К.

С37 Встраиваемые системы на основе Linux / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2017. – 360 с.: ил.

ISBN 978-5-97060-483-0

В книге подробно рассказано о том, как сконструировать встраиваемую Linux-систему из свободных программ с открытым исходным кодом, получив в результате надежный и эффективный продукт. Рассмотрены наборы инструментов, начальные загрузчики, ядро Linux и конфигурирование корневой файловой системы. Показано, как работать с системами сборки Buildroot и Yocto Project. Описаны процессы, потоки и управление памятью. Не обделены вниманием вопросы отладки и оптимизации платформы, а также выполнение приложений реального времени.

Издание рассчитано на разработчиков программного обеспечения на платформе Linux и системных программистов, уже знакомых со встраиваемыми системами. Предполагаются знание основ языка C и опыт системного программирования.

УДК 004.453/.453: 004.451.9Linux
ББК 32.972.1

Copyright © Packt Publishing 2016. First published in the English language under the title 'Mastering Embedded Linux Programming (9781784392536)'.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-1-78439-253-6 (анг.)
 ISBN 978-5-97060-483-0 (рус.)

Copyright © 2015 Packt Publishing
 © Оформление, издание, перевод, ДМК Пресс, 2017

Содержание

Предисловие	15
Об авторе	16
О рецензентах.....	17
Вступление	19
 Глава 1. Приступая к работе	 25
Выбор правильной операционной системы.....	26
Игроки	27
Жизненный цикл проекта.....	28
Четыре составные части встраиваемой Linux-системы	29
Программное обеспечение с открытым исходным кодом	30
Лицензии	30
Оборудование для встраиваемых Linux-систем	31
Используемое оборудование.....	33
Плата BeagleBone Black	33
QEMU	34
Используемое программное обеспечение	35
Резюме	35
 Глава 2. О наборах инструментов	 36
Что такое набор инструментов?.....	36
Типы наборов инструментов – платформенные и перекрестные	38
Архитектура процессора	39
Выбор библиотеки C.....	40
Получение набора инструментов	42
Сборка набора инструментов с помощью crosstool-NG	43
Установка crosstool-NG.....	43
Выбор набора инструментов.....	44
Анатомия набора инструментов	46
Получение информации о кросс-компиляторе.....	46
sysroot, библиотека и файлы-заголовки	48
Другие элементы набора инструментов.....	48
Компоненты библиотеки C	49
Статическая и динамическая компоновка с библиотеками.....	50
Статические библиотеки.....	50
Разделяемые библиотеки.....	51

Искусство кросс-компиляции	53
Простые make-файлы	54
Autotools.....	54
Конфигурирование пакета	57
Проблемы кросс-компиляции	58
Резюме	58

Глава 3. Все о начальных загрузчиках60

Что делает начальный загрузчик?.....	60
Последовательность начальной загрузки.....	61
Этап 1: код в ПЗУ	62
Этап 2: SPL.....	63
Этап 3: TPL	63
Загрузка из UEFI-прошивки	64
Переход от начального загрузчика к ядру.....	65
Введение в деревья устройств	66
Основные сведения о деревьях устройств	66
Свойство reg	67
Указатели на описатели и прерывания	68
Включаемые файлы деревьев устройств	69
Компиляция дерева устройств.....	71
Выбор начального загрузчика	71
U-Boot.....	72
Сборка U-Boot	72
Установка U-Boot	73
Работа с U-Boot	74
Загрузка Linux	78
Kconfig и U-Boot	79
Сборка и тестирование	82
Режим Сапсан	82
Varebox.....	82
Получение Varebox.....	83
Сборка Varebox.....	83
Резюме	84

Глава 4. Портирование и конфигурирование ядра86

Что делает ядро?.....	86
Выбор ядра	88
Цикл разработки ядра	88
Стабильные и долгосрочные версии	89
Поддержка со стороны производителя	90
Сборка ядра.....	90
Получение исходного кода	90

О конфигурировании ядра	91
Использование переменной LOCALVERSION для идентификации ядра.....	95
Модули ядра	96
Компиляция.....	96
Компиляция образа ядра.....	97
Компиляция деревьев устройств	99
Компиляция модулей.....	99
Удаление артефактов сборки	99
Загрузка ядра.....	100
BeagleBone Black.....	100
QEMU	100
Паника ядра	101
Подготовка пользовательского пространства	102
Сообщения ядра	102
Командная строка ядра.....	103
Портирование Linux на новую плату.....	104
С деревом устройств	104
Без дерева устройств	105
Дополнительная литература.....	107
Резюме	107

Глава 5. Построение корневой файловой системы 109

Что должно быть в корневой файловой системе?	109
Структура каталогов.....	111
Каталог технологической подготовки	111
Программы в корневой файловой системе	114
Программа init	114
Оболочка.....	115
Утилиты.....	115
BusyBox спешит на помощь!.....	115
ToyBox – альтернатива BusyBox	117
Библиотеки для корневой файловой системы	118
Уменьшение размера путем удаления таблицы символов	119
Узлы устройств.....	119
Файловые системы proc и sysfs	120
Монтирование файловых систем.....	121
Модули ядра	122
Перенос корневой файловой системы на целевое устройство.....	122
Создание загрузочного ram-диска.....	123
Автономный ram-диск.....	123
Загрузка ram-диска	124
Встраивание ram-диска в формате срjо в образ ядра	125
Старый формат initrd	126

Программа init.....	126
Конфигурирование учетных записей пользователей.....	127
Добавление учетных записей пользователей в корневую файловую систему.....	129
Запуск процесса-демона.....	129
Улучшенный способ управления узлами устройств.....	129
Пример использования devtmpfs.....	130
Пример использования mdev.....	130
А так ли плохи статические узлы устройств?.....	131
Конфигурирование сети.....	131
Сетевые компоненты для glibc.....	132
Создание образов файловой системы с помощью таблиц устройств.....	133
Копирование корневой файловой системы на карту SD.....	134
Монтирование корневой файловой системы по NFS.....	134
Тестирование в эмуляторе QEMU.....	135
Тестирование с платой BeagleBone Black.....	136
Проблемы с правами доступа к файлам.....	136
Загрузка ядра по протоколу TFTP.....	137
Дополнительная литература.....	138
Резюме.....	138

Глава 6. Выбор системы сборки 139

Довольно самопальных встраиваемых систем.....	139
Системы сборки.....	139
Форматы пакетов и менеджеры пакетов.....	141
Buildroot.....	141
История.....	142
Стабильные версии и поддержка.....	142
Установка.....	142
Конфигурирование.....	143
Выполнение.....	144
Создание специального BSP-пакета.....	145
Добавление своего кода.....	146
Соответствие лицензионным требованиям.....	148
Yocto Project.....	149
История.....	149
Стабильные версии и поддержка.....	150
Установка Yocto Project.....	151
Конфигурирование.....	151
Сборка.....	152
Выполнение.....	153
Слои.....	154
Настройка образов с помощью local.conf.....	159
Рецепт создания образа.....	159

Создание SDK.....	160
Контроль лицензий.....	162
Дополнительная литература.....	162
Резюме	162

Глава 7. Выбор стратегии хранения 164

Типы запоминающих устройств	164
Флэш-память типа NOR	165
NAND-память.....	166
Управляемая флэш-память	168
Доступ к флэш-памяти из начального загрузчика.....	169
U-Boot и флэш-память типа NOR	170
U-Boot и флэш-память типа NAND.....	170
U-Boot и карты MMC, SD и eMMC.....	170
Доступ к флэш-памяти из Linux.....	170
Устройства на основе технологии памяти.....	171
Драйвер блочного устройства MMC	176
Файловые системы для флэш-памяти.....	177
Уровень флэш-преобразования	177
Файловые системы для флэш-памяти типа NOR и NAND.....	178
JFFS2.....	178
YAFFS2	181
UBI и UBIFS	182
Файловые системы для управляемой флэш-памяти.....	186
Flashbench.....	187
Discard и TRIM.....	188
Ext4	189
F2FS.....	190
FAT16/32	190
Сжатые неизменяемые файловые системы	191
squashfs.....	191
Временные файловые системы	192
Превращение корневой файловой системы в неизменяемую.....	193
Варианты выбора файловой системы	194
Обновление в месте эксплуатации	194
Степень детализации: файл, пакет или образ?.....	195
Атомарное обновление образа.....	196
Дополнительная литература.....	197
Резюме	198

Глава 8. Введение в драйверы устройств 199

Роль драйверов устройств	199
Символьные устройства	200

Блочные устройства	202
Сетевые устройства	203
Получение информации о драйверах на этапе выполнения	205
Получение информации из sysfs	207
Устройства: /sys/devices	207
Драйверы: /sys/class	208
Блочные устройства: /sys/block	209
Поиск подходящего драйвера устройства	209
Драйверы устройств в пользовательском пространстве	210
GPIO	210
Светодиоды	213
Шина I2C	214
Шина SPI	216
Написание драйвера устройства	216
Проектирование интерфейса символьного устройства	216
Анатомия драйвера устройства	218
Загрузка модулей ядра	222
Определение конфигурации оборудования	222
Деревья устройств	223
Платформенные данные	223
Связывание оборудования с драйверами	224
Дополнительная литература	226
Резюме	226

Глава 9. Инициализация системы – программа init 228

После того как ядро загрузилось	228
Введение в программы init	229
BusyBox init	229
Скрипты инициализации в Buildroot	231
System V init	231
inittab	233
Скрипты init.d	235
Добавление нового демона	235
Запуск и остановка служб	236
systemd	237
Сборка systemd в Yocto Project и Buildroot	237
Как systemd загружает систему	239
Добавление своей службы	240
Добавление сторожевого таймера	241
Применение во встраиваемых Linux-системах	242
Дополнительная литература	243
Резюме	243

Глава 10. Процессы и потоки	244
Процесс или поток?.....	244
Процессы.....	246
Создание нового процесса	246
Завершение процесса	247
Выполнение другой программы	249
Демоны	251
Межпроцессное взаимодействие.....	251
Потоки	256
Создание нового потока	256
Завершение потока.....	258
Компиляция многопоточной программы.....	258
Межпроцессное взаимодействие.....	258
Мьютексы	259
Изменение условий.....	259
Разбиение проблемы на части.....	260
Планирование	262
Справедливость и детерминированность.....	262
Политики с разделением времени	263
Политики реального времени.....	264
Выбор политики.....	265
Выбор приоритета реального времени.....	266
Дополнительная литература.....	266
Резюме	267
 Глава 11. Управление памятью	 268
Основы виртуальной памяти.....	268
Структура памяти ядра.....	269
Сколько памяти потребляет ядро?	270
Структура памяти в пользовательском пространстве	272
Карта памяти процесса	274
Подкачка	275
Выгрузка страниц в сжатую память (zram).....	275
Отображение памяти с помощью mmap.....	276
Использование mmap для выделения частной памяти	276
Использование mmap для разделения памяти	276
Использование mmap для доступа к памяти устройства.....	277
Сколько памяти потребляет мое приложение?.....	277
Потребление памяти на уровне процесса.....	278
Использование top и ps.....	278
Использование smem.....	279
Другие инструменты.....	281

Обнаружение утечек памяти.....	281
mtrace	281
Valgrind	282
Нехватка памяти	283
Дополнительная литература.....	285
Резюме	285
Глава 12. Отладка в GDB	287
Отладчик GNU.....	287
Подготовка к отладке.....	287
Отладка приложений в GDB	288
Удаленная отладка с помощью gdbserver	289
Настройка Yocto Project	290
Настройка Buildroot	290
Начало отладки.....	290
Подключение GDB к gdbserver.....	290
Задание sysroot	291
Командные файлы GDB.....	292
Обзор команд GDB	292
Выполнение до точки прерывания	293
Отладка разделяемых библиотек	294
Yocto Project.....	294
Buildroot.....	295
Другие библиотеки.....	295
Своевременная отладка	295
Отладка разветвлений и потоков	296
Core-файлы	296
Использование GDB для анализа core-файлов	298
Пользовательские интерфейсы к GDB	298
Терминальный пользовательский интерфейс	299
Отладчик DDD.....	299
Eclipse.....	300
Отладка кода ядра.....	301
Отладка кода ядра в kgdb.....	301
Пример сеанса отладки.....	302
Отладка на ранних стадиях.....	304
Отладка модулей.....	304
Отладка кода ядра в kdb.....	305
Сообщения об ошибках ядра	306
Сохранение сообщения об ошибке ядра.....	307
Дополнительная литература.....	308
Резюме	309

Глава 13. Профилирование и трассировка 310

Эффект наблюдателя.....	310
Таблицы символов и флаги компиляции	311
Приступая к профилированию	311
Профилирование с помощью top	312
Профилировщик для бедных	313
Применение perf	314
Конфигурирование ядра для работы с perf.....	314
Сборка perf в Yocto Project.....	315
Сборка perf в Buildroot.....	315
Профилирование с помощью perf.....	315
Графы вызовов	317
perf annotate	318
Другие профилировщики: OProfile и gprof	319
Трассировка событий.....	321
Введение в Ftrace	321
Подготовка к работе с Ftrace	322
Динамический режим Ftrace и фильтры трассировки	324
События трассировки.....	325
Использование LTTng.....	326
LTTng и Yocto Project	327
LTTng и Buildroot	327
Применение LTTng для трассировки ядра	327
Использование Valgrind для профилирования приложений	330
Callgrind.....	330
Helgrind	330
Использование strace для показа системных вызовов.....	331
Резюме	333

Глава 14. Программирование в режиме реального времени..... 335

Что такое реальное время?	335
Определение источников недетерминированности	337
Задержки планирования	339
Вытеснение в ядре	340
Ядро Linux реального времени (PREEMPT_RT).....	340
Потоковые обработчики прерываний.....	341
Вытесняемые блокировки ядра.....	343
Получение заплат PREEMPT_RT	344
Yocto Project и PREEMPT_RT	344
Таймеры высокого разрешения.....	345
Предотвращение страничных отказов в приложении реального времени	345

14 ❖ Содержание

Экранирование прерываний	346
Измерение задержек планирования	347
cyclictest	347
Ftrace	350
Комбинирование cyclictest и Ftrace	352
Дополнительная литература	352
Резюме	353
Предметный указатель	354